CS10720 Problems and Solutions

Thomas Jansen

Today: Introduction
Representing Data

January 28^{th}

Plans for Today

Welcome!

What?	CS10720 Problems and Solutions
When & Where?	Monday 5–6pm here (Physics Main) Thursday 1–2pm here (Physics Main) + Practicals
Who?	Thomas Jansen
Contact?	forum in `http://blackboard.aber.ac.uk` office E45 (Llandinam building) (office hours each Monday 1–3pm & flexible on appointment) email t.jansen@aber.ac.uk (PGP key available) `http://users.aber.ac.uk/thj10`
Assessment?	in-class test (50 minutes, 30%) 'portfolio' (continuous, 40%) exam (2 hours; 30%)

Practicals

Idea	putting contents into practice mostly by programming in C a bit by using a Turing machine simulator and by doing a few exercises
When & Where?	either Tuesday 9–11am in LL-B23 or Wednesday 9–11am in LL-B23 beginning next week (i. e., 2nd or 3rd of February)

Sad Fact practicals require a bit of preparation to be effective

- know what was done in the lectures the week before
- know how to look up details from those lectures (and earlier ones)

About Assessment

Component 1	in-class test weight 30% 50 minutes, in lecture slot 07/03/2016
Component 2	'portfolio' (implemented as blog on Blackboard) weight 40% reproduce and apply lecture contents regularly as instructed (contents partly from the practicals)
Component 3	written exam weight 30% use sample questions now to prepare (not only at the end) discussion of solutions in forum in Blackboard (i. e., no solutions from me)

Assessment Component 2: Portfolio

Idea

- continuously create a 'log' of what you learn in CS107
- create an ideal basis for revision when exam time comes
- be prepared for the exam better than usual
- get 40% of marks for just a bit of weekly work
  (as opposed to one big assignment)

Implementation

- implemented as blog on Blackboard
- empty individual blogs are (hopefully) visible

Work this week (by TOMORROW, 11am!)

- check that blog is accessible and inform me immediately if not
- create brief summary of lecture contents (only non-organisational parts) based on the lecture today

What does 'brief' mean?

- as short as you can while naming each significant topic
- containing sufficient details to understand what the topic is about and how things are done

Material

- slides (available on Blackboard)
- lecture notes (available on Blackboard)
- exercises (available on Blackboard)
  as we go (highly recommended for exam preparation)
- invitation to self-assessment (available on Blackboard)
  to give you a vague indication how you are doing
- books (list available on Aspire)

- L. Null/J. Lobur (2014): *The Essentials of Computer Organization and Architecture*. 4th ed. Jones & Barlett Learning. Available in the library (QA76.9.C643.N9)
- D. A. Patterson/J. L. Hennessy (2012): *Computer Organization and Design. The Hardware/Software Interface*. 4th ed. Morgan Kaufmann. 3rd ed. available online: `http://site.ebrary.com/lib/aber/docDetail.action?docID=10382827`
- A. S. Tanenbaum (2006): *Structured Computer Organization*. 5th ed. Pearson Prentice Hall. available in the library (QA76.9.C643.T1)
- R. L. Graham/D. E. Knuth/O. Patashnik (1994): *Concrete Mathematics*. 2nd ed. Addison Wesley. available in the library (QA39.2.G7)
- M. A. Vine (2002): *C Programming for the Absolute Beginner*. Ebrary. Available online: `http://site.ebrary.com/lib/aber/Doc?id=10065758`
- G. W. Lecky-Thompson (2007): *Just Enough C/C++ Programming*. Ebrary. Available online: `http://site.ebrary.com/lib/aber/Doc?id=10228169`
- T. H. Cormen/C. E. Leiserson/R. L. Rivest/C. Stein (2001): *Introduction to Algorithms*. 2nd ed. MIT Press. available in the library (QA76.6.C8)
- C. Bishop/J. MacCormick (2012): *Nine Algorithms that Changed the Future.* Princeton University Press.

Making Lectures Interactive

Ideal lecture as a dialogue

Reality I'll be talking much more than you
introducing new material, explaining things

Please,

- feel free to interrupt any time
- let me know immediately if I am too fast or too slow
- ask if something is not clear

Method to guarantee minimum level of 'instant' feedback
onlineTED (`onlineted.com`) (instead of Qwizdom)
Drawback Requires you to have Internet access here.

`http://onlineted.com`

Overview: Things to Look Forward to

Grand Theme What are the foundations and highlights of CS?

- fundamentals of computer science (4 lectures)
- analysing algorithms (2 lectures)
- sorting (2 lectures)
- matrices and arrays (2 lectures)
- recursion and induction (5 lectures)
- computability (2 lectures)
- CS highlights (3 lectures)

Why would you care? because you want to be a computer scientist

Module contents

- things every computer scientist should know
- important fundamental concepts
- limits of what computers can do
- example of computer science that touches everybody's life

Representing Data

Reminder We restrict ourselves to digital computers
not analog computers (and also not quantum computers)

Consequences

- We can represent movies as silent movies and accompanying sound.
- We can represent silent movies as finite sequences of pictures.
- We can represent pictures as finite collections of pixels.
- We can represent pixels as numbers (coordinates, colour).
- We can represent sound as finite sequences of amplitudes.
- We can represent amplitudes as numbers.
- We can represent programs as texts (e. g., Java, C, . . .).
- We can represent texts as numbers (e. g., ASCII, Unicode).
- We can represent everything using numbers.

How do we represent numbers?

Representing Numbers (Stuff You've 'Always' Known)

Background a bit philosophy that's really important
distinguish between entity (thing) and its representation
(Remember Kant's 'thing-in-itself vs phenomenon'
or Plato's cave allegory)

Example different representations of the same thing
4 four IV vier ●●●●

You already know

- decimal representation

  e. g.,

  $3842 = 3 \cdot 1000 + 8 \cdot 100 + 4 \cdot 10 + 2 \cdot 1 = 3 \cdot 10^3 + 8 \cdot 10^2 + 4 \cdot 10^1 + 2 \cdot 10^0$

  base 10

- mixed representation

  e. g., $1{:}45 = 1 \cdot 60 + 45$ (= 105 in case you care)

  base 60 (and base 10 for numbers below 60)

Representing Numbers (Stuff You Know From CS101)

- binary representation
  e. g., $101010 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5$
  $= 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 8 + 0 \cdot 16 + 1 \cdot 32$
  $= 42$ in case you care

  base 2
- hexadecimal representation
  e. g., $A9F2 = 2 \cdot 16^0 + \underbrace{F}_{15} \cdot 16^1 + 9 \cdot 16^2 + \underbrace{A}_{10} \cdot 16^3$
  $= 2 \cdot 1 + 15 \cdot 16 + 9 \cdot 256 + 10 \cdot 4096$
  $= 43506$ in case you care

  base 16

Notation here we use $(\text{number})_{\text{base}}$
if the base is not clear from context
for example

$$(101)_2 = 2^0 + 2^2 = 1 + 4 = 5$$
$$(101)_{16} = 16^0 + 16^2 = 1 + 256 = 257$$
$$(101)_{10} = 10^0 + 10^2 = 1 + 100 = 101$$

Integers in Binary Representation

Observation for dealing with binary digital computers
binary representation particularly useful and important

What about negative numbers?

How do we normally represent negative numbers?

Remember using a minus sign $-$
e. g., -4, -17

Observation a minus sign is a different symbol
$\rightsquigarrow$ no longer binary representation
bad for digital computers

Goal find good binary representation for negative integers
i. e., representation using only 0 and 1
still allowing to distinguish negative and positive integers

Fundamentals for Binary Integer Representation

From now on numbers represented with a fixed length
(using representation length $l = 6$ in all examples)
What about with too short numbers?
For example, how do I represent 15 with 6 digits?
using leading 0s
for example representing 15 in decimal with 6 digits as ‘000015’

Remark makes sense because it is close to computer hardware
usually using 32 bits or 64 bits as representation lengths

We know how to represent non-negative integers in binary
and use this to find ways of presenting negative integers in binary

Fact There are countless possibilities.
We cover four different important standards.

Integers in Binary Representation: Signed Magnitude

Remember standard binary representation
for example $(11)_{10}$ represented as 001011

Simple idea use left most bit to signal sign (called sign bit)
decide 0 signals non-negative number ('$0 \mathrel{\widehat{=}} +$')
1 signals negative number ('$1 \mathrel{\widehat{=}} -$')
Why? because $(-1)^0 = 1$ and $(-1)^1 = -1$

Example $(11)_{10}$ represented as 001011
$(-11)_{10}$ represented as 101011

How is 0 represented?

Observation 000000 represents $(-1)^0 \cdot 0 = 1 \cdot 0 = 0$
100000 represents $(-1)^1 \cdot 0 = -1 \cdot 0 = 0$

Remark two different representations for 0 are unpleasant
because '$+0 = -0$' but $000000 \neq 100000$ on bit-level
makes comparison harder for computers to perform

Example: Conversion Decimal $\rightarrow$ Signed Magnitude

Example Convert -19 to Signed Magnitude Rep. with 6 bits

Observation -19 is negative, so sign bit is 1

Conversion of 19 into binary
(different methods available, here using repeated division)
$19/2 = 9$ R 1 thus, least significant bit is 1
$9/2 = 4$ R 1 thus, next bit is 1
$4/2 = 2$ R 0 thus, next bit is 0
$2/2 = 1$ R 0 thus, next bit is 0
$1/2 = 0$ R 1 thus, next bit is 1
since we are at 0 all remaining bits are 0
Result 10011

Result 110011 (because we use 6 bits as length)

Example: Signed Magnitude $\rightarrow$ Decimal

Example Convert 100110 from Signed Magnitude Rep.

Observation sign bit is 1, thus number is negative

Conversion 110 into decimal
$(110)_2 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 = 2 + 4 = 6$
Result 6

Result -6

The Extreme Values

What is the largest number you can represent with $l = 6$ bits in signed magnitude representation?

Representation 011111

Value $1 + 2 + 4 + 8 + 16 = 31$

And in general, for arbitrary l?

Representation $\underbrace{0111\cdots11}_{0 \text{ and } l-1 \text{ 1-bits}}$

Value $1 + 2 + 4 + 8 + \cdots + 2^{l-2} = 2^{l-1} - 1$

What is the smallest number you can represent with $l = 6$ bits in signed magnitude representation?

Representation 111111

Value $-(1 + 2 + 4 + 8 + 16) = -31$

And in general, for arbitrary l?

Representation $\underbrace{1111\cdots11}_{l \text{ 1-bits}}$

Value $-(1 + 2 + 4 + 8 + \cdots + 2^{l-2}) = -\left(2^{l-1} - 1\right) = -2^{l-1} + 1$

Summary & Take Home Message

Things to remember

- Blackboard: slides, lecture notes, discussion forum, 'portfolio'
- 'portfolio' to be done by you by 11am each Friday
- office hours Monday or by appointment (t.jansen@aber.ac.uk)
- CS107 covers foundations of CS
- representing non-negative integers: decimal, binary, hexadecimal
- representing integers: signed magnitude

Take Home Message

- Knowing the basics is important.
- Numbers can be represented in different formats.
- Standards are important.

Lecture feedback `http://onlineted.com`